1)l0’ algoritmo di round robin e un algoritmo di scheduling di tipo prehemptive (capace di mettere in pausa l’esecuzione di un processo) ogni processo viene eseguito per un quanto ti tempo (time slice) prima di mettere in pausa l’esecuzione e passare all prossimo processo questo concede una divisione delle risorse più o meno uniforme tra i vari processi.

Se il time slice e troppop lungo però l’algoritmo diventa un FCFS first come first served dato che il processo terminerà la sua esecuzione prima della fine del quanto di tempo

Se il time slice e troppo corto però si sprecherà un sacco di tempo di overhead nel eseguire le operazioni di context switching (che richiedono molto tempo di io per salvare lo stato del processo)

2)

Negli algoritmi di scheduling si può introdurre una priorita che aiuta a decidere quale processo sarà eseguito per primo (la priorità e rappresentata come un valore da 0 a 5 dove in alcuni sistemi operativi il valore più alto e quello con maggiore priorità mentre in altri il valore più basso, 0 e la maggiore priorità processi di maggiore priorità come processi di sistema possono usare un algoritmo di scheduling divero) cio ci concede di eseguire processi più importanti come quelli di sistema ciò però puo causare un problema cioè la starvation dove un processo di priorità più alta impedisce l’esecuzione di un altro per questo si usa un algoritmo di aging che conta da quanto tempo e in attesa un processo e gli alza la priorità in base a ciò.

3)

Per il programatore la memoria dovrebbe essere infinita poco costosa e veloce ciò non e possibile quindi bisogna arrivare a un compromesso utilizzando tipi di memoria diversi

Velocità e capienza sono infatti inversamente proporzionali tra di loro (più aumenta la velocità più diminuisce la capacita) e direttamente proporzionati al costo

La gerarchia e strutturata cosi:

Registri

Cache (di vari livelli)

Ram

Memoria di massa:  
{

A stato solido ssd

Disco Rigido hdd 1-10tb

Nastro magnetico 1-100tb

}  
la mem